White Paper Report

Report ID: 100935

Application Number: HD5110110

Project Director: Stefaan Van Liefferinge (svlieffe@uga.edu)

Institution: University of Georgia

Reporting Period: 9/1/2010-1/31/2011

Report Due: 4/30/2011

Date Submitted: 5/4/2011

Artificial Intelligence for Architectural Discourse
White Paper*

Stefaan Van Liefferinge, Charles Hollingsworth, Rebecca A. Smith, Michael A. Covington, Walter D. Potter

The University of Georgia

Introduction

This paper introduces a work in progress on a new research topic for artificial intelligence. It presents the first stages of research that investigates the capacities of AI to understand the architectural rules that define historic architecture: specifically those of Gothic cathedrals. Gothic architecture has its own logic. It follows rules that are defined by physical and structural constraints as well as by the style of the architecture. This project will analyze specialized descriptions of Gothic architecture and translate them into computer code. On the long run this will result in an intelligent computational model, a program that can understand the structure of a Gothic cathedral and reason about its architecture. While earlier research has pointed towards the importance of architecture as a future domain of application for AI, for various reasons, little has been achieved in recent years. This paper introduces the background of this project, sketches a small knowledge base illustrating the approach, and describes the impacts that this research can have on the relation between architecture and computers.

Overall Method

The limits prescribed by architectural design make built structures a good subject for AI experimentation. It is thus not surprising that a number of projects have brought architecture within the scope of AI. For example, architectural historians have indicated how patterns in the language of architecture can be approached as if it were code; engineers have investigated the possibilities of AI for developing computer generated models of architecture [1,2,3]. These general studies underscore the affinity between AI and architecture. However, a system that codifies and that is capable of understanding the rules that regulate built structures does not yet exist. One of the reasons for this lacuna is that, up to now, research has focused directly on the buildings themselves without addressing a mediating system, a translation that maps a build structure into a language.

* A slightly modified version of this white paper will be presented at the 2011 International Conference on Artificial Intelligence (ICAI) and will be published in the conference proceedings. This research benefited from the generous support of a Digital Humanities Start-Up Level I Grant from the National Endowment for the Humanities and of a University of Georgia Research Foundation Grant.

This research project involves a method that utilizes a translation of architecture into a logical system. More precisely, it takes advantage of the existence of architectural descriptions in the literature, descriptions that are expressed in a natural language, and those that follow specific rules. Indeed, the conventions which architecture generally obeys have defined a language, a structured and logical means of describing architecture. Architectural historians commonly manipulate this tool to communicate their analysis of buildings. These descriptions reflect the structure and rules of architecture, and, conversely, the conventions that architectural texts follow are well adapted to describing built structures.

Thus, instead of searching for a direct mapping between actual buildings and an intelligent computer system, our research investigates the application of AI technologies for understanding written description of buildings. For this task a subset of architecture, Gothic cathedrals, was chosen. This subset is well adapted because Gothic cathedrals are known for being organized following recurrent patterns. Their internal organization is valuable and serves as a heuristic to codify the logic of their architecture. In our research, methods commonly used to describe these buildings are analyzed and categorized, and the rules that are extracted are translated into algorithms developed for AI. NLP is addressed to understand the specific language that architectural historians use, and a KR of the language and logic of actual architecture is developed.

While describing architecture is a practice in obeying conventions, it is nonetheless a human activity that combines presupposed knowledge, intelligence, and creativity. These factors put this research project well beyond coding or modeling. It searches to catch the processes involved in a specific discipline - architectural history - and to bring these within the reach of computational logic. Ultimately, we would like to implement a system that can automatically translate between English and a computer, and which can reason about any description of architecture. However, the translation of architectural descriptions into formal logic is no simple task. For this, at this stage, we are experimenting with the translation of a small subset of architecture, that of a column, into a declarative language and testing how a computer can operate and interactively reason with this subset.

A Minimal Case Study

Because the project aims at identifying knowledge presupposed in descriptions of architecture, knowledge representation is of central importance to our project. Architectural descriptions must be translated into a formal, machine-readable format that can be used for responding to queries and drawing inferences. We decided to render these descriptions as source code for the logic programming language Prolog, because the Prolog interpreter already incorporates question answering and inference. Additionally, much work has already been done on translating from natural language to Prolog [4,5].

We chose first to devise a knowledge base for describing Gothic columns, with the intention of expanding this later to include other architectural features.

Columns were chosen because they are fairly simple structures. Certain features, such as the base, capital, and shaft, are found in all columns, while others, such as neckings and plinths, are not always present (Fig. 1). Furthermore, the features in a column are arranged very simply, one on top of the other, and the order does not vary. We thus needed to be able to represent concepts such as "Every column has a shaft" and "If a column has a necking, then the necking is immediately below the capital." Finally, we needed to account for the fact that many architectural features are repeated, so that one might encounter statements such as "Every vaulting unit has four columns." For this last requirement, we made extensive use of skolemization, as described in [5].

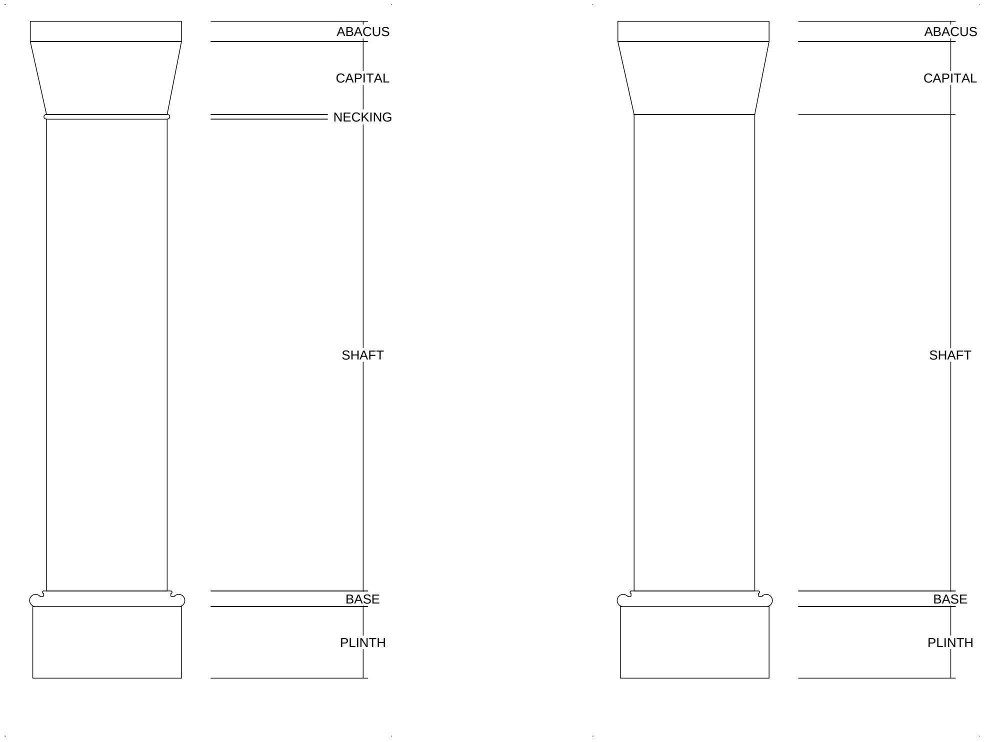

Fig. 1. Column, with and without necking

The Prolog translation of the assertion that "Every column has a shaft" is in Figure 2.

The Prolog term `[shaft_inst(1),X]` is a Skolem function, a way of providing a unique identifier for each shaft that is a function of the name of the

```
shaft([shaft_inst(1),X]) :-
        column(X).

has(X, [shaft_inst(1),X]) :-
        column(X).
```

Fig. 2. Prolog translation of “Every column has a shaft.”

column of which it is a component. In other words, we are asserting two things: for each column `X`, there exists a unique shaft named `[shaft_inst(1),X]`; and column `X` has that shaft as one of its components. These identifiers are admittedly unwieldy, but should only be used internally by our program: in the finished product, the end user should never have to type them as part of a query. The listing in Figure 3 illustrates how to say “The necking is immediately above the shaft,” while bearing in mind that not all columns will have a necking.

```
immediately_above(X,Y) :-
        necking(X),
        shaft(Y),
        has(ParticularColumn , X),
        has(ParticularColumn , Y).
```

Fig. 3. Prolog translation of “If a column has a necking, it is immediately above the shaft.”

Essentially this states that, if some particular column has both a necking and a shaft, the necking is immediately above the shaft. Elsewhere we deal with columns without neckings, in which case it is the capital which is immediately above the shaft. Our knowledge base even exhibits some non-monotonic reasoning: if we don’t explicitly tell it that a column has a necking, it will assume that the capital tops the shaft; if we then assert that it does have a necking, the features will be represented in their proper order. At present, here is a very basic set of queries which the user can make to our knowledge base. To get a list of constituent parts of a column named column1, for example, the user can type `has(column1, X).` into the Prolog interpreter. For an exhaustive list of the components of vaulting unit v1 and their various sub-components, type `has(v1,X), has(X,Y)`. To get a list of all the components below the capital in `column1`, type `has(column1, X), capital(X), above(X, Y).`

While this simple example already captures some essential skills needed for architectural description, we will expand this knowledge base to deal with more and more sophisticated concepts. Eventually, a natural language front end will take the place of the Prolog query interface, making the system more usable, and paving the way for eventually being able to handle more typical architectural texts.

Future Impact

Both architectural history and artificial intelligence will benefit from the combination of technology and traditional methods of building analysis. It will allow architectural historians to understand better how we write architectural descriptions, will be a beneficial study tool for students and professionals alike, and will eventually be able to give us accurate visuals of buildings. Engineers working in artificial intelligence can expect from this project an entirely new scope for natural language processing, and to apply artificial intelligence to a previously little-explored field.

Writing for architectural history is a complex and often subjective process. The number of assumptions made when writing a description of a building, particularly when writing about Gothic architecture, is vast. A program that could "read" and analyze architectural descriptions would be entirely new. Such a program would be able to detect trends among various essays and treatises that scholars are often unaware exist. These trends could be as simple as the order in which a building elevation is described, to something as complex as the nature of the geometrical pattern from which the plan is derived. The discovery of patterns within written text could further the study of natural language processing by highlighting how humans describe buildings. It could show how we translate the experience of a building into written description – and possibly underscore the limitations within writing. Previously unknown peculiarities within individual buildings could be found and studied by experts, without affect from the voice of a particular author or the subjective analysis of future scholars.

One of the long-term goals of this project is to create a program that can generate accurate models that integrate 3D images of buildings, expert knowledge of architectural historians, and archaeological data. This will have a profound effect on the study of architectural history and provide a fascinating field of exploration for artificial intelligence. For example, by using written sources, the system will be able to create a building design or plan that is historically more valid than that of a human scholar drawing by CAD or by hand. Applications of this type of image creation program should also be of interest to engineers applying AI methods in the virtual game design industry. Engineers could create buildings and cities that were previously limited to the human imagination. The program could use historical written accounts to understand better what the original author actually meant to describe, even if the account is mistaken or lacking. This will allow us to create models and imagery of buildings that exist only in descriptive accounts.

Often drawings of buildings that no longer exist or may have never existed are inaccurate and misleading. This project opens a door to the generation in the future of accurate and detailed elevations, floor plans, and virtual tours. Such models could, for example, inform preservationists about previous structural interventions in buildings. This would enable more accurate repairs, which would lead to the continuing survival of many of the world's treasures. The cooperation between these two seemingly remote fields will be mutually beneficial, and likely foreshadows the coming of new interdisciplinary studies – combining artificial intelligence with both architectural history, and other fields in the humanities and sciences.

References

1. Yong Liu, Congfu Xu, Qiong Zhang, and Yunhe Pan. The smart architect: Scalable ontology-based modeling of ancient chinese architectures. *IEEE Intelligent Systems*, 23:49–56, 2008.
2. William J. Mitchell. *The logic of architecture: Design, computation, and cognition.* MIT Press, Cambridge, MA, USA, 1990.
3. Pascal Müller, Peter Wonka, Simon Haegler, Andreas Ulmer, and Luc Van Gool. Procedural modeling of buildings. *ACM Trans. Graph.*, 25:614–623, July 2006.
4. Patrick Blackburn and Johan Bos. *Representation and inference for natural language: A first course in computational semantics.* CSLI Publications, Stanford, CA, USA, 1999.
5. Michael A. Covington, Donald Nute, Nora Schmitz, and David Goodman. *From English to Prolog via discourse representation theory.* Technical report, Advanced Computational Methods Center, The University of Georgia, 1988. *http://www.ai.uga.edu/ftplib/ai-reports/ai010024.pdf* (available as of November 2010).